Security Risks of Suspicious Data reduced by a Novel Model

Gullipalli Suneetha

NIT Calicut M.Tech (CSIS), Alumni of JNTUV(IT). Email: suneetha21095@gmail.com

Introduction:

Modern operating systems are notable for being open, allowing us to download data or run programmes from any Internet source, reputable or not, without hesitation. When we wish to use this data or codes and maintain the system safe at the same time, a paradox arises.

Even after decades of research and experience, this issue is still seen as a major obstacle. In order to lessen the security concerns brought on by this dubious data or programmes for open operating systems, this study provides a revolutionary dynamic defensive model (DDM). The elements enable DDM to provide complete, dynamic, and real-time security protection throughout the operating system's entire life cycle.

Dynamic Defense Model:

A high-level security defense abstraction with four essential parts is called the "Dynamic Defense Model": 1. Marking labels dynamically, 2. Tracking labels dynamically, 3. Modulating labels dynamically, 4. Run-time regulation.

First of all, we cannot completely trust them given all the questionable data and codes. As a result, we must give them a distinctive label to set them apart from reliable information or codes.

Second, when the OS is functioning, suspect and sensitive data are consumed and spread across the system, and suspect processes produce child processes. We need to monitor where this data flow, how much data was contaminated, and how these shady activities impact the system via dynamic label tracking.

Thirdly, the original label may not adequately describe the suspicious processes' current security station after real-time execution of the suspicious processes was seen. According to the suspect processes' most recent activity, the label should be dynamically modified.

Last but not least, we shouldn't let any data or programmes spread or operate freely in the system without any controls.

Data Label Marking:

DDM typically offers three different types of labels: risk, cap, and sens.

Risk label: As soon as they enter the system, we should identify all the suspect data andcodes that originate from Internet sources with specific names so that we can distinguish them from the reliable data and codes.

Cap label: The cap label is intended to stand for the most fundamental or minimal capabilities (permissions or privileges) that the suspicious code should be granted in order to carry out its intended function.

Sens label: The sens label is designed to represent how important the sensitive data or resources are.

Label tracking:

We define the high level rule of our label tracking as:

- For the risk label, this rule can be specified as:

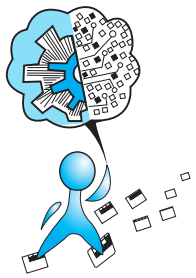
  $A \sim B \Rightarrow B.risk = max(A.risk, B.risk)$

  Label tracking is the same for both risk and sens labels.
- For the cap_o label, this rule can be specified as:

  $A \sim B \Rightarrow B.cap\_o = (A.cap\_o \cup B.cap\_o)$

Dynamic Label Modulating:

DDM modifies the label in three different ways: automatically incrementally, automatically decrementally, and Evaluative modulation.



The rule for automatic incremental modulation of the risk label can be described as:risk_new = risk_old + f (action_ill)

Similarly, the rule for automatic incremental modulation of sens and cap_o label can be described as: sens_new = sens_old + f (action_sen)

cap_o_new = cap_o_old + f (action_sen)

The rule for evaluative modulation can be described as: label_new = f (actions)

Runtime Controlling:

We cannot let any data or codes to spread or operate at will in the system without any controls. To avoid any potential harm to the system or the loss of confidential information, we should stop harmful acts in accordance with security regulations.

The following are examples of high-level runtime regulating rules: For the risk label, this rule can be specified as: running is permitted ⇐⇒ risk < threshold_kill

According to this rule, a process can continue to run if and only if the value of its risk label is lower than the process-killing threshold. For the risk label and sens label, this rule can be specified as: access is permitted ⇐⇒ risk + sens < threshold_sys

According to this rule, a process with a risk label can only access resources or sensitive data if the sum of the risk and sens values is below the system threshold. For the cap_s and the cap_o label, this rule can be specified as: access is permitted ⇐⇒ cap_o ⊆ cap_s

According to this rule, a process with the label cap s can only access resources or sensitive data with the label cap_o if and only if cap_o's value is a subset of cap_s.

Conclusion:

When we wish to employ dubious data or code from an Internet source that we can't completely trust while maintaining the operating system's security, it might be quite difficult. A unique dynamic defensive model is presented by the author as a solution to this issue, reducing the security threats posed by suspicious data or codes.

Reference:

[1] Reducing Security Risks of Suspicious Data and Codes Through a Novel Dynamic Defense Model Zezhi Wu, Xingyuan Chen, Zhi Yang and Xuehui Du.

About the Author

Gullipalli Suneetha is currently pursuing her Masters in the stream of Computer Science and Engineering with a specialisation in Information Security at the National Institute of Technology Calicut. Her interest lies in data and information security. Also, she is working on UAVs and their security, as drone and related attacks are becoming more dangerous nowadays.